DECIDABILITY OF REGULAR LANGUAGE GENUS COMPUTATION

Guillaume Bonfante, Florian Deloup

DECIDABILITY OF REGULAR LANGUAGE GENUS COMPUTATION

GUILLAUME BONFANTE, FLORIAN DELOUP

Abstract. The article continues the study of the genus of regular languages that the authors introduced in a 2012 paper. Let L be a regular language. In order to understand the genus $g(L)$ of L, we introduce the topological size of $|L|_{\text{top}}$ to be the minimal size of all finite deterministic automata of genus $g(L)$ computing L. We show that the minimal finite deterministic automaton of a regular language can be arbitrary far away from a finite deterministic automaton realizing the minimal genus and computing the same language, both in terms of the difference of genera and in terms of the difference in size. We show that the topological size $|L|_{\text{top}}$ can grow at least exponentially in size $|L|$. We conjecture the genus of every regular language to be computable. This conjecture implies in particular that the planarity of a regular language is decidable, a question asked in 1978 by R.V. Book and A.K. Chandra. We prove here the conjecture for a fairly generic class of regular languages having no short cycles.

Contents

1. Introduction

Regular languages form a robust and well-studied class of languages: they are recognized by finite deterministic automata (DFA), as well as various formalisms such as Monadic Second-Order logic, finite monoids, regular expressions. Traditionally, the canonical measure of the complexity of a regular language is given by the number of states of its minimal deterministic automaton.

In this paper, we study an alternative measure of language complexity, with a more topological flavor. We will be interested in the topological complexity of underlying graph structures of deterministic automata recognizing the language. Recall that the genus of an oriented surface Σ is the maximum number of mutually disjoint simple closed curves $C_1, \ldots, C_g \subset \Sigma$ such that the complement $\Sigma - (C_1 \cup \cdots \cup C_g)$ remains connected. This yields a natural notion of genus of a graph: a graph has genus n if it is embeddable in a surface of genus n but cannot be embedded in a surface of strictly smaller genus.

This definition was used in [BD13] to define the genus of a regular language L as the minimal genus among underlying graphs of deterministic automata recognizing L. In particular, L has genus 0 if and only if it can be recognized by a planar deterministic automaton. Here we provide new hierarchies of regular languages based on the genus, including for regular languages on two letters (Theorem 2).

One of the main questions is the computability of the genus of a regular language (Conjecture 1 below). This conjecture implies the decidability of the planarity of a regular language – a question raised in 1978 by R.V. Book and A.K. Chandra [BC76]. In this paper, we prove the conjecture for the class of regular languages having no short cycle (Theorem 6).

The complexity of the computation of the genus is reflected on the cost of extra states needed to build a deterministic automaton of minimal genus. We show that the number of states required may be exponential in the size of the minimal automaton of the language (Theorem 4).

An approach to the computation of the genus of a regular language L consists in considering all possible underlying directed graphs of the automata computating the same language L. This leads to the notion of directed emulator of a graph. In several aspects the notion is both similar to and distinct from the classical notion of emulator of a graph (see, for instance, [Hli10] for background and a survey on a related open question in graph theory). The main result is that the existence of a directed emulator of genus g of the underlying directed graph of the minimal automaton of a regular language is equivalent to the existence of a deterministic automaton of genus g computing the same language (Theorem 8).

Plan of the paper. Section 2 provides background, definitions of genus and topological sizes, examples (including hierarchy based on genus, exponential gap between size and topological size) and main results including the computability of the genus of a regular language for a class of regular languages without short cycles. Section 3 provides the set-up of directed emulators and the main equivalence between finding the genus of a regular language L and finding the minimal genus of directed emulators of the underlying directed graph of the minimal automaton for L. The proofs of the main results are collected in Section 4.

2. The genus and topological size of a regular language

2.1. Introductory examples.

The Myhill-Nerode theorem provides constructive existence and uniqueness of a DFA with minimal number of states recognizing a given regular language.

Definition 1. For each $k \geq 1$, we define the regular language on alphabet $\mathbb{Z}/k\mathbb{Z}$:

$$Z_k := \{a_1 a_2 \dots a_n \mid \sum_{i=1}^{n} a_i \equiv 0 \mod k\}.$$

It will be convenient to denote $Z_k^{a_1,\dots,a_r}$ the regular language obtained from Z_k by restriction to the subalphabet $\{a_1, \dots, a_r\} \subseteq \mathbb{Z}/k\mathbb{Z}$.

Example 1. The language $Z_5^{0,1,2}$. The figure below depicts the minimal automaton `A`. The transitions are of the form $i \xrightarrow{j} i + j \mod 5$.

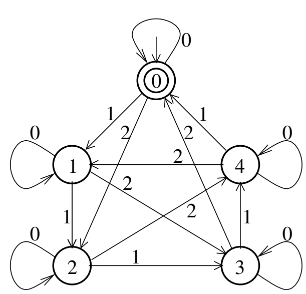


FIGURE 1. The minimal automaton for the language $Z_5^{0,1,2}$.

Since it contains the complete graph K_5, `A` is not planar. However, there exists a deterministic automaton with six states that is planar and computes the same language L:

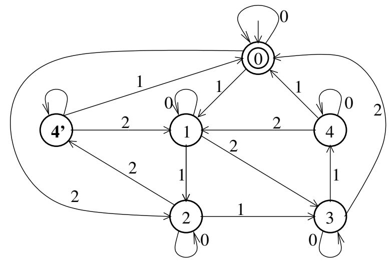


FIGURE 2. A planar automaton B computing L. Note that states 4 and $4'$ are equivalent.

In the previous example, adding just an extra state suffices to produce a planar equivalent automaton. The following example suggests that the general case may require many more states.

Example 2. The language Z_6. The figure below represents the minimal deterministic finite automaton A computing Z_6. It state space is $\mathbb{Z}/6\mathbb{Z}$ and its transitions are $i \xrightarrow{j} i + j \mod 6$, for all $i, j \in \mathbb{Z}/6\mathbb{Z}$.

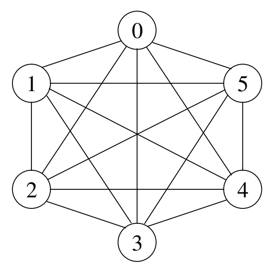


FIGURE 3. The minimal automaton of Z_6. For simplicity, the self-loop labelled 0 at each vertex is omitted and each edge represents two transitions in opposite directions.

There is no planar representation for A. (Since A has the complete graph K_6 as a minor, A is not planar.) However, there exists a deterministic automaton with 12 states that is planar and computes the same language L (Figure below). We regard the additional six states as the price to pay in order to simplify the topology of an embedding of the automaton into a surface. Since any 6-state automaton has an underlying graph which is a subgraph of Z_6, it follows easily that any language of size $|L| \leq 6$ can be represented by a planar finite deterministic automaton with at most 12 states.

2.2. **Genus-based hierarchies.** Let L be a regular language. In a previous article [BD13], we defined the *genus* $g(L)$ of L as the minimal genus of all embeddings of all finite deterministic automata recognizing L. A regular language is said to be *planar* if its genus is zero.

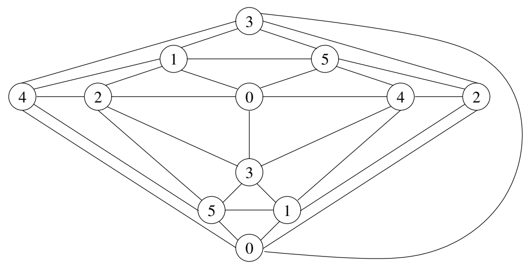


FIGURE 4. A deterministic automaton of minimal genus (planar) recognizing the same language Z_6 (with the same representation conventions as in the previous figure).

There are many nonplanar languages. A hierarchy of languages of strictly increasing genus is explicitly constructed in [BD13]. We produce other examples of such a hierarchy:

Theorem 1. *Let $k \geq 4$. The language $Z_{2k+1}^{1,2,\ldots,k}$ has genus $\lceil \frac{(2k-2)(2k-3)}{12} \rceil$. In particular, $g(Z_{2k+1}^{1,2,\ldots,k}) \underset{k\to+\infty}{\to} +\infty$.*

Note the closed formula for the genus. In general, the computation of the genus is nontrivial, as shall be explained further below. Note that the family of languages in Theorem 1 has an increasingly large alphabet. The examples provided in [BD13] have a fixed 4-letter (or more) alphabet. This left out regular languages on an alphabet with fewer letters, namely 2 or 3 letters. (Regular languages on a 1-letter alphabet are easily seen to be planar. See e.g. [BD13].) We shall prove here the following result.

Theorem 2. *There is a genus hierarchy of regular languages on only* 2-*letters: for any nonnegative integer $n \geq 0$, there exists a regular language L on a* 2-*letter alphabet such that $g(L) = n$.*

The result is constructive and explicit; it also implies the existence of a genus hierarchy of regular languages on any k-letter alphabet for $k \geq 2$ (since outgoing self-loops with arbitrary labels can always be added without affecting the genus).

2.3. **Genus, size and topological size.** Given a regular language L, we let $A_{\min}(L) = A_{\min}$ be the minimal deterministic automaton associated to L. The *size* $|L|_{\text{set}}$ of the language L is the size of the minimal deterministic automaton $A_{\min}$:

$$|L|_{\text{set}} = |A_{\min}|.$$

Definition 2. We define the *topological size* of L to be

$$|L|_{\text{top}} = \min\{|A| \mid L(A) = L,\ g(A) = g(L)\}$$

where the minimum is taken over all finite deterministic automata recognizing L of minimal genus.

By definition $|L|_{\mathrm{top}} \geq |L|_{\mathrm{set}}$ with equality if and only if the minimal automaton realizes the genus of L. From [BD13, §5] we know that the topological size is in general reached by several inequivalent deterministic automata. In light of the previous examples, a number of natural questions arise. What is the trade-off between size and genus ? Can a regular language be planar and its minimal automaton have an arbitrary high genus ? Indeed, the following result shows that the minimal DFA $A_{\min}$ can be arbitrarily far away from a DFA realizing the minimal genus.

Theorem 3. *There is a family of regular languages* $(L_n)_{n\geq 1}$ *such that*

$$|L_n|_{\mathrm{top}} - |L_n|_{\mathrm{set}} \underset{n\to\infty}{\to} +\infty \quad \text{and} \quad g(A_{\min}(L_n)) - g(L_n) \underset{n\to\infty}{\to} +\infty.$$

This can be pushed further. The following result shows that the topological size of L can grow at least exponentially in terms of (set-theoretic) size of L:

Theorem 4. *There is a family of planar regular languages* $(L_n)_{n\in\mathbb{N}}$ *and a positive number* $K > 1$ *such that*

$$|L_n|_{\mathrm{top}} = O(K^{|L_n|_{\mathrm{set}}}).$$

The idea of both constructions is to consider a sequence of planar languages L_n having increasingly high genus minimal automata $A_{\min}(L_n)$. In the first construction, planarity is ensured by a simple concatenation of the first example $Z_5^{0,1,2}$, known to be planar but whose minimal automaton has genus 1. In the second construction, planarity is ensured by spanning a tree for the language L_n while high genus of the minimal automaton is produced by means of a cascade of n directed $K_{5,5}$'s, completed by one initial state and one single final state.

The main tool to study the genus of a regular language consists in a lower bound in terms of its size. In order to state the result, we shall need a few definitions pertaining to the geometry of automata. Any automaton $\mathtt{A}$ gives rise to an undirected multigraph (by forgetting labels and orientations of transitions). Let $k \geq 1$. A *cycle of length* k in $\mathtt{A}$ is a closed walk in the underlying undirected multigraph of length k, considered up to circular permutation. Note that a cycle may or may not respect the orientation of the original transitions. A cycle of length 1 is also called a *loop* (or a *self-loop*, for emphasis). A cycle is *simple* if it is represented by a closed walk in which no edge is used more than once. In particular, a closed walk in which one edge is travelled twice in opposite directions does not induce a simple cycle.

It will be convenient to introduce some classes of regular languages that "do not have short cycles".

Definition 3. Let $j \geq 1$. A language L is said to be *without simple cycle of length* $\leq j$ if the minimal deterministic automaton $A_{\min}$ for L has no simple cycle of length $k \leq j$.

Example 3. The language $Z_5^{1,2}$ has no simple cycle of length ≤ 2. Indeed, the minimal automaton for $Z_5^{1,2}$ is the one depicted on Fig. 1 with all self-loops removed.

Remark 1. The notion depends on the alphabet. The language $L_1 = Z_3^1 = (\{1\}^3)^* = (111)^*$ has no simple cycle of length ≤ 2. The language $L_2 = (\{1,2\}^3)^*$ does have simple cycles of length 2.

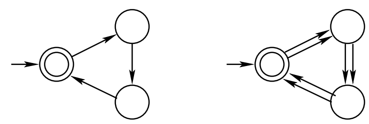


FIGURE 5. The minimal automata for L_1 and L_2. Note that they have the same underlying simple directed graph.

Theorem 5 (Genus estimate). *Let $m \geq 2$. Set* $j = \begin{cases} 3 & \textit{if } m \geq 4; \\ 4 & \textit{if } m = 3; \\ 5 & \textit{if } m = 2. \end{cases}$
If a regular language L on an m-letter alphabet has no simple cycle of length $\leq j-1$, then

$$1 + \frac{(j-2)m - j}{2j}|L|_{\mathrm{set}} \leq g(L) \leq 1 + \frac{(m-1)}{2}|L|_{\mathrm{set}}. \tag{1}$$

The upper bound is a direct consequence of Euler's formula. The crucial information consists in the lower bound. Theorem 5 generalizes that of [BD13, Theorem 8] and the proof goes along the same lines. See §5 for the detailed proof. One key observation is that if a nonsimple cycle (= closed walk) bounds a face in a minimal cellular embedding of a big enough graph, then its length is strictly greater than 4.

Conjecture 1. *The genus $g(L)$ of every regular language L is computable.*

Although the genus of a graph is computable, this conjecture is far from being obvious. Indeed, given a graph G and a nonnegative number g, there is a procedure, polynomial in time, that decides whether G embeds into a fixed surface of genus g and if is the case, determines an embedding (not uniquely determined). The known procedure is linear in the size of the graph (number of states) and doubly exponential in the genus g. However, this is not enough in order to say anything about the genus of a *language*: it is recognized by an infinity of DFAs and since the genus may be realized far away from the minimal DFA, it is not a priori clear where and when to stop. How far ? A priori according to Th. 3, it can be arbitrary far. In order to prove the conjecture, one needs a priori bounds that depend on the intrinsic complexity (ideally the size) of the language.

We prove a partial case of the conjecture above.

Definition 4. Keeping the notation $j = j(m)$ for $m \geq 2$ introduced in Theorem 5, we let $\mathscr{C}_j(m)$ denote the class of regular languages on m letters without simple cycles of length $\leq j-1$.

Theorem 6. *Let $m \geq 2$. For each $L \in \mathscr{C}_j(m)$, the topological size $|L|_{\mathrm{top}}$ and the genus $g(L)$ are computable.*

Corollary 6.1. *The planarity of a regular language $L \in \mathscr{C}_j(m)$ for $m \geq 2$ is decidable.*

Since regular languages are ordered by their genus, the following finiteness result is useful.

Theorem 7. *Let $m \geq 2$. If $\mathtt{A}$ is a deterministic finite automata $\mathtt{A}$ without simple cycles of length $\leq j(m)$, then $g(\mathtt{A}) \geq 2$. Furthermore, for each $g \geq 2$, there is a finite number of deterministic finite automata $\mathtt{A}$ without simple cycles of length $\leq j(m)$ such that $g(\mathtt{A}) = g(L(\mathtt{A}))$.*

Corollary 7.1. *Let $m \geq 2$. For any $L \in \mathscr{C}_j(m)$, $g(L) \geq 2$. Furthermore, for each $g \geq 2$, there is a finite number of regular languages $L \in \mathscr{C}_j(m)$ such that $g(L) = g$.*

A few comments may be useful. The hypotheses about the absence of small short cycles and the fixed size of the alphabet is essential. For instance, let $n, p \geq 3$ and consider the language on two letters

$$L_{n,p} = \{w \in \{0,1\}^* \mid |w|_0 = 0 \bmod n,\ |w|_1 = 0 \bmod p\}$$

(where $|w|_a$ denotes the number of occurrences of letter a in the word w) which can be regarded as the shuffle of Z_n^1 and $Z_p^{1'}$ [Sak03, p.65]. (One should distinguish the letter 1 of Z_n^1 and the letter $1'$ of $Z_p^{1'}$.) The minimal automaton for $L_{n,p}$ is obtained as the shuffle product of the minimal automata of Z_n^1 and Z_p^1 respectively. It is not hard to see that this automaton realizes the minimal genus for $L_{n,p}$ – which is 1. Clearly, by letting n and p vary, one obtains an infinite family of toric languages with two letters.

Clearly, for any n, p, the minimal automaton has simple cycles of length 4, so $L_{n,p} \notin \mathscr{C}_5(2)$. Another observation is that given a language $L \in \mathscr{C}_j(m)$, it is easy to build an infinite number of languages of the same genus $g(L)$, but of course the produced languages will have short simple cycles. For instance, if A denotes the alphabet of L and has at least two letters, then for any $k \geq 0$, $g(A^k \cdot L) = g(L)$. In this case, the composition $A \cdot L$ has one simple cycle of length 2 as is seen from the minimal automaton for L and adjoining a new initial state and transitions with labels in A from the new initial state to the old initial state.

3. Directed emulators

In this section, we give a graph-theoretical approach to the study of the genus of regular languages. By a digraph we mean a directed graph. A *morphism $G \to H$ between directed graphs* is a map $f : V_G \to V_H$ that

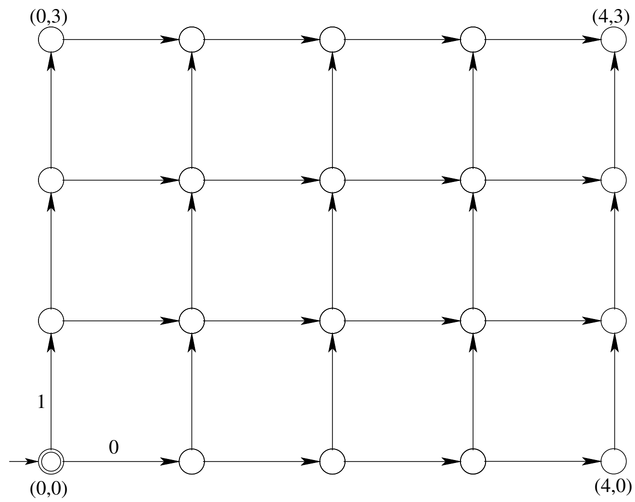


FIGURE 6. The minimal automaton for $L_{4,3}$ and its embedding in the torus. The states $(k, 0)$ and $(k, 3)$ $(0 \leq k \leq 4)$ and the states $(0, l)$ and $(4, l)$ $(0 \leq l \leq 3)$ are to be identified as well as the corresponding transitions so that there are exactly $12 = 4 \times 3$ states and $24 = 2 \times 12$ transitions.

preserves the adjacence relation, i.e., such that for any directed edge $e \in E_G$ from $\partial_0 e \in V_G$ to $\partial_1 e \in V_G$, there is a directed edge in E_H from $f(\partial_0 e) \in V_H$ to $f(\partial_1 e) \in V_H$. A *simple* digraph is a digraph whose directed edges form a set (rather than a multiset).

A *morphism* $\mathtt{A} \to \mathtt{B}$ *between automata* is a map $f : Q_{\mathtt{A}} \to Q_{\mathtt{B}}$ from the set of states of $\mathtt{A}$ to the set of states of $\mathtt{B}$ with the following properties:

(1) f sends the initial state of $\mathtt{A}$ to the initial state of $\mathtt{B}$;
(2) f sends the set of final states of $\mathtt{A}$ into the set of final states of $\mathtt{B}$;
(3) The following diagram commutes:

$$\begin{array}{ccc} Q_{\mathtt{A}} \times \mathcal{A} & \longrightarrow & Q_{\mathtt{A}} \\ {\scriptstyle f\times \mathrm{id}}\downarrow & & \downarrow{\scriptstyle f} \\ Q_{\mathtt{B}} \times \mathcal{A} & \longrightarrow & Q_{\mathtt{B}} \end{array}$$

where $\mathcal{A}$ denotes the alphabet and the horizontal maps are the transition maps of $\mathtt{A}$ and $\mathtt{B}$ respectively.

Deterministic finite automata with their morphisms form a category DFA_0. (See e.g., [Eil74, III.4].) We investigate more closely this category and the related category of directd graphs. For this we introduce a notion that was defined by M. Fellows in the context of undirected graphs.

Definition 5. Let $G = (V, E)$ be a digraph. We say that a digraph $G' = (V', E')$ is a *directed emulator* of G if there is a surjective map $\pi : V' \to V$

such that for all $(x, y) \in E$ and all $x' \in \pi^{-1}(x)$, there is $y' \in \pi^{-1}(y)$ such that $(x', y') \in E'$. Such a map π will be called a *directed emulator map*. In other words, if we regard the digraph as a simplicial 1-complex, a directed emulator map is a simplicial map mapping the outgoing edges from each vertex $x' \in V'$ surjectively onto the outgoing edges from the image vertex $x \in V$. We say that a digraph $G = (V, E)$ is a *directed amalgamation* of $G' = (V', E')$ if G' is a directed emulator of G.

Remark 2. Two distinct endpoints of an edge may happen to be sent to a single vertex (provided the local condition at the vertex is respected).

Remark 3. The deletion of a vertex (and all its outgoing and incoming edges) induces a directed emulator map.

Remark 4. The definition of a direct emulator is a weakening of the definition of a directed graph covering. A covering map maps the outgoing edges from each vertex $x' \in V'$ bijectively onto the outgoing edges from the image vertex $x \in V$. A graph covering map is a special kind of emulator map. An emulator map is a special kind of surjective simplicial map.

Remark 5. The digraph that consists of one vertex and no edge is the directed amalgamation of any nonempty digraph.

So far the definition of directed emulator makes sense in the category of directed (multi)graphs as well as in the category of simple digraphs. We shall now focus on the category of simple digraphs.

There is a forgetful functor $\widetilde{\mathscr{G}}$ from the category DFA_0 of deterministic finite automata on finite alphabets to the category DiGr of simple finite digraphs: $\widetilde{\mathscr{G}}$ forgets the labels on the transitions, the distinguished states and the (self)loops, and merges the multiple transitions between two identical ordered pairs of states into one transition.

Lemma 1. *The functor $\widetilde{\mathscr{G}}$ is full and preserves the genus of objects.*

In particular, a regular language L gives rise, via its minimal automaton $A_{\min}(L)$, to a simple digraph denoted $G(L)$.

Consider the category DFA:

- An object in DFA is a morphism in DFA_0, i.e. a morphism $\mathtt{A}' \to \mathtt{A}$ between complete deterministic finite automata.

- A morphism in DFA is a commutative diagram of automata morphisms:

$$\begin{array}{ccc} \mathtt{A}' & \longrightarrow & \mathtt{B}' \\ \downarrow & & \downarrow \\ \mathtt{A} & \longrightarrow & \mathtt{B} \end{array} \tag{2}$$

Consider the category DiEm of directed emulators ("di-emulators") defined as follows. An object in DiEm is a directed emulator map (i.e. a particular kind of morphism in DiGr). A morphism in DiEm is a commutative diagram

(3)
$$\begin{tikzcd} G' \arrow[r] \arrow[d, "\pi_G"'] & H' \arrow[d, "\pi_H"] \\ G \arrow[r] & H \end{tikzcd}$$

where the vertical maps are directed emulators and the horizontal maps are digraph morphisms.

Proposition 1. *The functor* $\widetilde{\mathscr{G}} : \mathrm{DFA}_0 \to \mathrm{DiGr}$ *induces a functor* $\mathscr{G} : \mathrm{DFA} \to \mathrm{DiEm}$.

Proof. We really only need to verify that the functor sends a morphism $p : \mathtt{A}' \to \mathtt{A}$ between automata to a directed emulator graph. Consider two distinct states $x, y \in \mathtt{A}$ such that $\delta_{\mathtt{A}}(x, l) = y$ for some letter l. Let a state $x' \in p^{-1}(x)$. By property (3) of the definition, $\delta_{\mathtt{A}'}(x', l)$ lies in $p_{\mathtt{A}}^{-1}(y)$. Therefore there is a transition from x' to some $y' \in p_A^{-1}(y)$ in $\mathtt{A}$. This implies that the induced map $\mathscr{G}(\mathtt{A}') \to \mathscr{G}(\mathtt{A})$ is a directed emulator map. $\square$

Definition 6. Let $\mathtt{A} \in \mathrm{DFA}_0$ be minimal. Let $\mathrm{DFA}(\mathtt{A})$ be the set of all $\mathtt{B} \in \mathrm{DFA}$ whose minimal automaton is $\mathtt{A}$.

Lemma 2. *Given* $\mathtt{A}_1, \mathtt{A}_2 \in \mathrm{DFA}(\mathtt{A}_{\min})$, *there exists* $\mathtt{A}_{12} \in \mathrm{DFA}$ *such that the following diagram is commutative*

$$\begin{tikzcd} & \mathtt{A}_{12} \arrow[dl, dotted, "p_{121}"'] \arrow[dr, dotted, "p_{122}"] \arrow[dd, dotted, "p_{12}"] & \\ \mathtt{A}_1 \arrow[dr, "p_1"'] & & \mathtt{A}_2 \arrow[dl, "p_2"] \\ & A_{\min} & \end{tikzcd}$$

Proof. Take $\mathtt{A}_{12}$ to be the fibered product of $\mathtt{A}_1$ and $\mathtt{A}_2$ over $p_1 \times p_2$. $\square$

Definition 7. Let G be a digraph. We denote $\mathrm{DiEm}(G)$ the set of directed emulators of G.

Lemma 3. *Given* $G_1, G_2 \in \mathrm{DiEm}(G)$, *there exists* $G_{12} \in \mathrm{DiEm}(G)$ *making the following diagram commute.*

$$\begin{tikzcd} & G_{12} \arrow[dl, dotted] \arrow[dr, dotted] \arrow[dd, dotted] & \\ G_1 \arrow[dr] & & G_2 \arrow[dl] \\ & G & \end{tikzcd}$$

where each map is a directed emulator map.

Proof. Apply the functor $\mathscr{G}$ to the diagram of the previous lemma. $\square$

Lemma 4. *Any two digraphs have a common directed emulator.*

Proof. They both are directed emulators of the digraph that consists of one vertex with no edge (Remark 5). $\square$

Definition 8. Let L be a regular language. The *underlying directed graph* $G(L)$ of L is the directed graph $\tilde{\mathscr{G}}(\mathtt{A}_{\min})$ associated to the minimal automaton $\mathtt{A}_{\min}(L)$ of A.

Theorem 8. *Let L be a regular language. The following assertions are equivalent:*

(1) *The language L has genus at most g.*
(2) *The associated digraph $G(L)$ has a directed emulator of genus at most g.*

Theorem 8 allows to translate questions about the genus of languages into questions about directed emulators of digraphs (and vice-versa).

Corollary 8.1. *If two languages L and L' have the same underlying directed graphs $G(L)$ and $G(L')$, then $g(L) = g(L')$.*

Corollary 8.2. *Let A be a deterministic automaton and L be the language computed by A. Let A' be a deterministic automaton obtained from A by the following operations:*

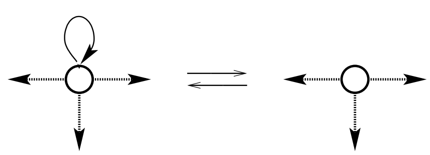


FIGURE 7. Adding or removing a self-loop

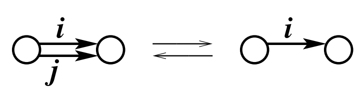


FIGURE 8. Adding or removing one extra transition between states

Let L' be the language computed by A'. Then $g(L) = g(L')$.

Caution needs to be exercised to apply this corollary since some of the operations (those adding transitions) do not preserve a priori determinism.

Proof. The operations do not affect the underlying directed graph of the respective minimal automata of A and A', so $G(L) = G(L')$. $\square$

4. The proofs

4.1. **Proof of Theorem 1 (A new explicit example of genus hierarchy with exact genus formula).** The language $Z_{2k+1}^{1,2,\ldots,k}$ is computed by the following automaton, denoted $A = A_{2k+1}^{1,2,\ldots,k}$. The set of states is $Q = \mathbb{Z}/(2k+1)\mathbb{Z}$, with initial and final state 0. The transitions are given by the rule $i \pmod{2k+1} \xrightarrow{j} i+j \pmod{2k+1}$ for $i \in Q$ and $j \in \{1, 2, \ldots, k\} \subset$

$\mathbb{Z}/(2k+1)\mathbb{Z}$. It is readily observed that A is the minimal automaton. The underlying unoriented multigraph is the complete graph K_{2k+1}. We verify two properties:

- K_{2k+1} has no self-loop and has no true cycle of length 2 (the minimal length of a true cycle is 3)[1].
- The cardinality of the alphabet is $k \geq 4$.

According to [BD13, Th. 8], $g(Z_{2k+1}^{1,2,\ldots,k}) \geq 1 + \frac{(k-3)(2k+1)}{6}$. To prove that this lower bound for the genus is actually an equality, we notice that the genus of the minimal automaton provides an upper bound. So

$$1+\frac{(k-3)(2k+1)}{6} \leq g(Z_{2k+1}^{1,2,\ldots,k}) \leq g(A) = g(K_{2k+1}) = \left\lceil \frac{(2k-2)(2k-3)}{12} \right\rceil.$$

The last equality is the exact formula for the genus of the complete graph on $2k+1$ vertices. It remains to observe that the ceiling function of the lower bound is exactly the upper bound. This is the desired result.

4.2. **Proof of Theorem 2 (Existence of a genus hierarchy for languages on a 2-letter alphabet).** Let $A = \mathbb{Z}/2\mathbb{Z}$ be the alphabet. For $k \geq 5$, consider the finite deterministic automaton $\mathtt{A}_k$ defined as follows. The set of states is $Q_k = \mathbb{Z}/6\mathbb{Z} \times \mathbb{Z}/k\mathbb{Z}$. The transitions are

$$(i,j) \xrightarrow{0} (i+1,j), \quad (i,j) \xrightarrow{1} (2i, j+1).$$

Pick the state $(0,0)$ as the initial and unique final state. See Fig. 9 for a picture of the automaton $\mathtt{A}_k$.

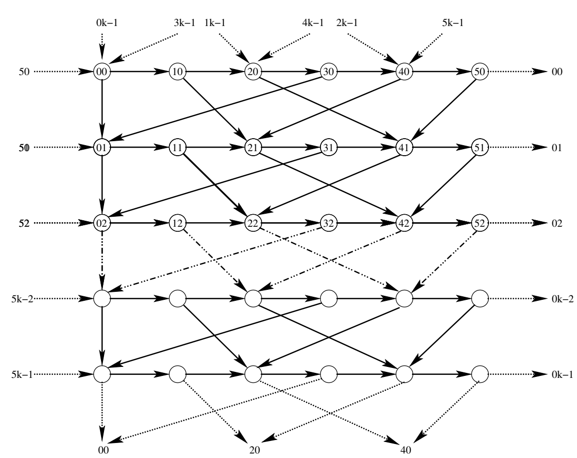


FIGURE 9. The automaton A_k. Horizontal arrows are labelled by 0; non-horizontal arrows are labelled by 1. For better readability, some states are repeated more than once.

[1]Here it is crucial that $2k+1$ is odd, otherwise one would have multiple edges and hence cycles of length 2.

It is easily seen that $\mathtt{A}_k$ is deterministic, complete and minimal. It is readily verified that $\mathtt{A}_k$ has no simple cycle of length less than or equal to 4. Therefore Theorem 5 applies: the language L_k recognized by $\mathtt{A}_k$ has genus $g(L) \geq 1 + \frac{3k}{20}$. This implies the desired result.

4.3. **Proof of Theorem 3 (Genus versus size).** We shall construct a family of regular languages L_n with the prescribed property. With a slight modification of the example, one produces infinitely many families with the desired property. The idea is to use a planar regular language whose minimal automaton has genus 1. We take the regular language L considered in Example 1 with A being the minimal automaton recognizing L. It has notoriously genus 1. However, L is planar and $|L|_{\text{top}} = 6$. We form the automaton A_n that consists of a necklace of n copies of A as depicted below. There are n top states denoted $0_0, 0_1, \ldots, 0_{n-1}$ for convenience. The state 0_0 is the only entry and exit state.

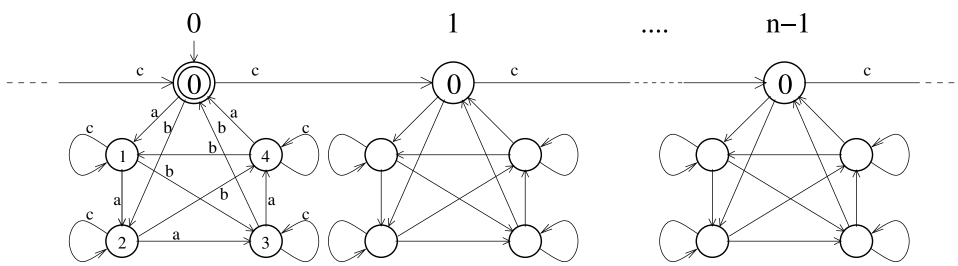


FIGURE 10. The necklace of n copies of automaton A.

Let $L_n = L(A_n)$. We make the following claims:
1. The language L_n is planar (has genus 0) for any $n \geq 1$.
2. $|L_n|_{\text{set}} = |A_n| = 5n$.
3. $|L_n|_{\text{top}} = |B_n| = 6n$.
Assuming these claims, we have

$$|L_n|_{\text{top}} - |L_n|_{\text{set}} = n \underset{n\to\infty}{\to} \infty$$

as desired. We now prove each claim.

Proof of claim 1. We have already seen in Example 1 that there is a planar six-state DFA B such that $L(B) = L$. Form a necklace B_n that consists of n copies of B in exactly the same way A_n was formed from A. This operation preserves planarity and L_n is the language recognized by B_n. Hence L_n is planar.

Proof of claim 2. We prove that A_n is the minimal automaton for L_n. Consider two distinct states s, s' in A_n. Let us say that s lies in the k-th copy of A and s' lies in the l-th copy of A with $0 \leq k, l \leq 4$ and that s is the i-th state and s' is the j-th state with $0 \leq i < j \leq 4$ (see Fig. 10). The word $a^{n-i}c^{n-k}$ lies in L_s; it lies in $L_{s'}$ if and only if $i = j$ and $k = l$. Hence s and s' are not equivalent.

Proof of claim 3. For $n = 1$, L_n is the language L computed by A. By claim 1, L is planar; since the minimal automaton A computing L is not planar, we have $|L|_{\text{top}} \geq |A| + 1 = 6$. Thus the planar representation B of claim 1 that has only six states has the minimal number of states: $|L|_{\text{top}} = |B| = 6$.

Let us prove the result for $n \geq 1$. Clearly $|B_n| = 6n$. We have to show that B_n has the minimal number of states among all deterministic automata of genus 0 computing the language L_n. Consider a deterministic automaton C_n of genus 0 computing L_n. We have to prove that $|C_n| \geq 6n$. Since A_n is the minimal automaton, C_n projects canonically onto A_n. Consider the set $S = \{s_1, \ldots, s_r\}$ of states in A_n that have a preimage that consists of at least two states in C_n. We claim that for each copy of A in A_n, there must be at least one state among $s_1, \ldots, s_r$ that lies in it. For if it is not the case, then C_n contains one full copy of A. Since A has genus 1, C_n cannot be planar, contrary to the assumption. Since there are n copies of A in the necklace A_n, it follows that S has cardinality at least n. Hence $|C_n| \geq |A_n| + |S| = 5n + n = 6n$.

We are left proving the assertion about the genus. By Claim 1, $g(L_n) = 0$ for any $n \geq 1$. By Claim 2, $A_{\min}(L_n) = A_n$. The graph A_n has a block decomposition into n blocks, each of which is a copy of A. Since the genus is additive with respect to blocks [Bat62, Th. 1],

$$g(A_n) = n\ g(A) = n \cdot 1 = n.$$

This completes the proof.

4.4. **Proof of Theorem 4 (Planar regular languages with exponential topological size).** On the alphabet $\mathbb{Z}/5\mathbb{Z}$, given $n \geq 0$, let us consider the automaton $A_n = (Q_n, i_n, F_n, \delta_n)$ defined as follows. The set of states is $Q_n = \mathbb{Z}/5\mathbb{Z} \times \{0, \cdots, n\} \cup \{p_0, \top, \bot\}$. The initial state is p_0, there is a unique final state $\top$. For all $a, b \in \mathbb{Z}/5\mathbb{Z}$, let $\delta_n(p_0, a) = (a, 0)$, $\delta_n((a, n), a) = \top$, if $a \neq b$, $\delta_n((a, n), b) = \bot$ and for $j < n$, $\delta_n((a, j), b) = (a + b, j + 1)$. Its corresponding language is $L_n = \{a_0 \cdots a_{n+1} \mid \sum_{i=0,n} a_i = a_{n+1}\}$.

It is straightforward that all states of A_n are accessible and that A_n is minimal, its states being non equivalent. The language L_n is finite, thus planar. Indeed, one may span the complete tree of depth $n+2$ to describe the language which has thus topological size smaller than 5^{n+2}. Let us suppose that $B_n = (R_n, j_n, G_n, \eta_n)$ is a minimal planar automaton recognizing L_n. Without loss of generality, we can suppose that its states have the shape (s, t) with $s \in Q_n$ and $t \in T$, that is $\pi : (s, t) \mapsto s$ defines the projection on the minimal automaton.

We qualify states of the shape (a, j, t) with $j < n$ to be internal states. For any internal state $s = (a, j, t)$, the transition function $\eta_n(s, \cdot) : \mathbb{Z}/5\mathbb{Z} \to R_n$ is injective, because $\delta_n = \pi \circ \eta_n$ is injective. Explicitly, for any $b \neq c \in \mathbb{Z}/5\mathbb{Z}$, we have $\eta_n(s, b) \neq \eta_n(s, c)$.

Let $G_n = \widetilde{\mathscr{G}}(B_n)$ be the underlying (planar) graph of B_n. Given $j \in \{0, \ldots, n-1\}$, let S_j be the subgraph of G_n where any vertices outside $\mathbb{Z}/5\mathbb{Z} \times \{j, j+1\} \times T$ have been removed with their incoming and outcoming edges. Being a subgraph of G_n, the graph S_j is planar. We denote K (respectively M) the set of states of B_n of the shape (a, j, t) (resp. $(a, j+1, t)$) and $k = |K|$ (resp. $m = |M|$).

Any state $s \in K$ is internal. We have seen above that $\eta_n(s, \cdot)$ is injective. Thus, there are exactly 5 outgoing edges from state s, each of which pointing to a different state. Two partial conclusions. First, let e be the number of edges in S_j, we have $e = 5k$. Second, there are no bigons in S_j: none of the patterns $s \to s' \to s$ or $s \to s' \leftarrow s$ can happen.

Let f be the number of faces in S_j. Euler's formula for planar graphs applied in S_j gives us $k + m + f = 5k + 2$, that we can rewrite: :

$$m + f = 4k + 2. \tag{4}$$

Let f_i be the number of i-gon in S_j. Thus, $f = f_1 + f_2 + \cdots$. Observe that due to the definition of B_n, there are neither simple odd polygons (that is a $2i+1$-gon for $i \in \mathbb{N}$), neither bigons as justified above. Thus, $f = f_4 + f_6 + \cdots$. A simple counting argument shows that $2 \times e = 4f_4 + 6f_6 + \cdots = 10k$. In other words, $\frac{5k}{2} = f_4 + \frac{6}{4} f_6 + \cdots \geq f_4 + f_6 + \cdots = f$. By relation (4), we get

$$m = 4k + 2 - f \geq \frac{3k}{2} + 2 \geq \frac{3k}{2} \tag{5}$$

Take $\Phi = 3/2$. Denote by N_j the states in layer j, that is of the shape (a, j, t), and by n_j the cardinal of N_j. By induction on $j \geq 0$, we prove $n_j \geq 5 \times (3/2)^j$ for $j \leq n$. For the base case, observe that there are at least 5 states in each layer (there are 5 in the minimal automaton). The induction step is a direct consequence of the inequality (5). The result follows.

4.5. **Proof of Theorem 5 (Genus estimate).** We need to prove the stated lower bound. Set

$$A(j) = \sum_{k \geq j} \frac{k(m-1) - 2m}{4m} f_k, \quad B(j) = \sum_{k \geq j} k\, f_k.$$

Then

$$A(j) \geq \left(\frac{m-1}{4m} - \frac{1}{2j} \right) B(j).$$

Let $\mathtt{A}$ be a complete minimal genus finite deterministic automaton recognizing L. By [BD13, Th. 5], $g(\mathtt{A}) = 1 + A(1)$. By hypothesis, $A_{\min}$ has no simple cycle of length less or equal to $j-1$. It follows from Lemma 5 that $\mathtt{A}$ has no simple cycle of length less or equal to $j-1$. Consider a minimal embedding (hence cellular) of $\mathtt{A}$ into a genus $g(L)$ oriented closed surface Σ. Consider now a face f in Σ. If the length of the face is less or equal to 4, then any cycle c of $\mathtt{A}$ bounding f must be simple. We deduce that there is no face of length less than $j-1$: $f_1 = \cdots = f_{j-1} = 0$. Hence

$$\begin{aligned} g(\mathtt{A}) = 1 + A(j) &\geq 1 + \left(\frac{m-1}{4m} - \frac{1}{2j}\right) B(j) \\ &\geq 1 + \left(\frac{m-1}{4m} - \frac{1}{2j}\right) B(1) \\ &= 1 + \left(\frac{m-1}{4m} - \frac{1}{2j}\right) 2m|\mathtt{A}| \end{aligned}$$

Therefore

$$g(\mathtt{A}) \geq 1 + \frac{(j-2)m - j}{2j}|\mathtt{A}|. \tag{6}$$

Since $|\mathtt{A}| \geq |A_{\min}| = |L|_{\text{set}}$, we deduce the desired result.

Remark 6. The inequality (6) holds for any complete minimal genus deterministic automaton recognizing L under the hypotheses of Theorem 5. It is in general sharper than the lower bound of the theorem.

Remark 7. There are planar simple graphs supporting nonsimple bounding closed walks (see Fig. 11): the complement in the 2-sphere of each of these embedded graphs is an open cell whose boundary is a nonsimple closed walk in the graph. It is left to the reader to verify that if one of the graphs below is the underlying simple graph of a 2-letter automaton $\mathtt{A}$, then the underlying multigraph of $\mathtt{A}$ has a *simple* cycle of length less or equal to 4.

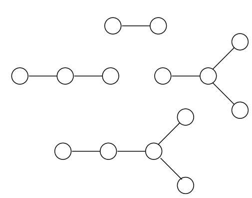


FIGURE 11. Planar graphs supporting nonsimple bounding cycles.

4.6. **Proof of Theorem 6 (Computability of genus).** Let $\mathtt{A}$ be a deterministic finite automaton such that $L(\mathtt{A}) = L$ and $g(\mathtt{A}) = g(L)$. Let $A_{\min}$ be the minimal deterministic automaton for L. By Lemma 5, since $A_{\min}$ has no simple cycles of length $\leq j-1$, neither has $\mathtt{A}$. According to the inequality (6) and [BD13, Prop. 2],

$$1 + \left(\frac{(j-2)m-j}{2j}\right)|\mathtt{A}| \leq g(A) = g(L) \leq g(A_{\min}) \leq 1 + \left(\frac{m-1}{2}\right)|A_{\min}|.$$

Since $|L|_{\text{set}} = |A_{\min}| \leq |\mathtt{A}|$, we have

$$1 + \left(\frac{(j-2)m-j}{2j}\right)|L|_{\text{set}} \leq g(\mathtt{A}) \leq 1 + \left(\frac{m-1}{2}\right)|L|_{\text{set}}.$$

The set

$$E = \left\{ n \in \mathbb{N} \mid |L|_{\text{set}} \leq n,\ 1 + \left(\frac{(j-2)m - j}{2j}\ n \right) \leq g(A_{\min}) \right\}$$

is finite and contains $|L|_{\text{top}}$. Let $n \in E$. There is only a finite number of DFAs of fixed size n, hence a finite number of DFAs of size n and computing L. Therefore the set

$$F = \{ \mathtt{A} \in \mathrm{DFA}(A_{\min}) \mid L(\mathtt{A}) = L,\ |\mathtt{A}| \in E \}$$

is finite and contains every deterministic finite automaton computing L of minimal genus. Now for each individual automaton $\mathtt{A} \in F$, its genus is computable (computation of the genus of a graph). The minimum of the finite list of genera thus computed is the genus of L.

More generally, given a finite graph $\mathtt{A}$, there is a known algorithm to construct all embeddings of $\mathtt{A}$ into a surface of minimal genus. It follows from the argument above that there is an algorithm to construct every deterministic finite automaton $\mathtt{A}$ such that $L(\mathtt{A}) = L$ and $g(\mathtt{A}) = g(L)$. In particular, there is only a finite number of them $\mathtt{A}_1, \ldots, \mathtt{A}_r$. In particular, one can compute $|L|_{\text{top}} = \min\{|\mathtt{A}_1|, \ldots, |\mathtt{A}_r|\}$. This completes the proof.

4.7. **Proof of Theorem 7 (Finiteness of minimal fixed genus automata).** Let $m \geq 2$. According to (6) (see Remark 6), $g(\mathtt{A}) > 1$ for any deterministic finite automaton $\mathtt{A}$ without simple cycles of length $\leq j-1$. It follows that there is no deterministic finite automaton $\mathtt{A}$ without simple cycles of length $\leq j-1$ such that $g(\mathtt{A}) \in \{0, 1\}$. In particular, there is no language $L \in \mathscr{C}_j(m)$ of genus 0 or 1.

Let $g \geq 2$. Let $\mathtt{A}$ be any deterministic finite automaton $\mathtt{A}$ without simple cycles of length $\leq j-1$, such that $g(\mathtt{A}) = g(L(\mathtt{A})) = g$. According to (6),

$$1 + \frac{(j-2)m - j}{2j}\,|\mathtt{A}| \leq g.$$

Since the set of sizes

$$\left\{ n \geq 0 \mid 1 + \frac{(j-2)m - j}{2j}\, n \leq g \right\}$$

is finite and since there is a finite number of automata with prescribed size n and prescribed alphabet size m, the claimed result follows.

4.8. **Proof of Theorem 8 (Equivalence with genus g directed emulators).** Suppose that $g(L) \leq g$. Then there exists $\mathtt{A} \in \mathrm{DFA}$ such that $L(\mathtt{A}) = L$ and $g(\mathtt{A}) \leq g$. By Prop. 1, $\widetilde{\mathscr{G}}(\mathtt{A})$ is a directed emulator over $\mathscr{G}(A_{\min})$. By Lemma 1, $g(\widetilde{\mathscr{G}}(\mathtt{A})) = g(\mathtt{A}) \leq g$. Therefore, $G(L) = \mathscr{G}(A_{\min}(L))$ has a directed emulator of genus $\leq g$.

Conversely, suppose $G(L) \in \mathcal{P}_g$. This says that there is a directed emulator map $\pi : G' \to G(L)$ where G' is a simple digraph of genus at most g. By Lemma 1, this morphism lifts to a morphism $\mathtt{A} \to A_{\min}(L)$ between automata, with $g(\mathtt{A}) = g(G') \leq g$. Hence $g(L) \leq g(\mathtt{A}) \leq g$. This proves the equivalence (1) $\Longleftrightarrow$ (2).

4.9. **Cycles and directed emulators.**

Lemma 5. *Let $k \geq 1$. Assume that a directed graph G has no simple cycle of length less than or equal to k. Then neither has any directed emulator $\tilde{G}$ over G.*

Proof. Suppose that $\tilde{G}$ has a simple cycle c' of length $l \leq k$. Its image in G is a closed path c of length l. The closed path c admits a decomposition into a product of simple cycles, each of which has length less than or equal to $l \leq k$. ☐

Lemma 6. *The property (P_k) for a deterministic automaton $\mathtt{A}$ to have no simple cycle of length $l \leq k$ is a property of the language $L(\mathtt{A})$.*

Proof. Let $A_{\min}$ be the minimal automaton of $L(\mathtt{A})$. Set $G = \tilde{\mathscr{G}}(A_{\min})$: the digraph $\tilde{G} = \tilde{\mathscr{G}}(\mathtt{A})$ is a directed emulator of G (by Prop. 1). ☐

Lemma 7. *Let $\pi : \tilde{G} \to G$ be a directed emulator map. Let c be a simple cycle of length k in G. Then $\pi^{-1}(c)$ contains a simple cycle of length a multiple of k in $\tilde{G}$.*

Proof. Let $c = v_1 \cdots v_n$ where the v_i's denote the vertices. Choose an arbitrary lift v_1' of v_1. Since π is a directed emulator map, each edge $e_i = v_i v_{i+1}$ of c has a lift starting at any lift v_i' of v_i. Lifting each edge of c in this fashion, we obtain a path $c' = v_1' \cdots v_n' v_1''$ whose initial and final vertex lie in the same fibre: $\pi(v_1') = \pi(v_1'') = v_1$. If the initial and final vertices of c' coincide, we stop and c' is a simple cycle. Otherwise, starting again with v_1'' as a lift of v_1, we continue the process of lifting edges until we reach a first vertex w that has already been reached. This implies that there is a path in $\tilde{G}$, which lifts c, starts and ends at w. Among the closed path lifting c, let $\tilde{c}$ be closed path of minimal length with this property. Suppose that $\tilde{c}$ is not simple. Then there is an edge e' of $\tilde{c}$ which is repeated, which implies that there are two pairs of vertices that are repeated, contradicting that $\tilde{c}$ has minimal length. Thus $\tilde{c}$ is simple. Since moreover $\tilde{c}$ covers c, the length of $\tilde{c}$ is a multiple of that of c. ☐

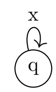
In what follows, we suppose that the languages we are talking about are subsets of $\{a, b\}^*$. Id est, m = 2: all automata are defined on the two letter alphabet.

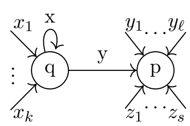
Proposition 2. *Let $\mu : G \hookrightarrow H$ be an injective graph morphism and $H' \to H$ such that H' is a direct emulator of H. Then, the projection of the pullback $\pi : G \times_H H' \to G$ is a directed emulator of G.*

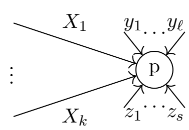
Proof. Direct consequence of the definitions. ☐

With the proposition above, we can state that if we find a planar direct emulator of H, then, any subgraph G of H has a planar direct emulator.

Theorem 9. *Let $m = 2$. All languages with minimal automaton $\mathtt{A}_{min}$ verifying $|\mathtt{A}_{min}| \leq 7$ is planar. Among languages whose minimal automaton size is 8, there is exactly one language (up to isomorphism) with genus one. It is $Z_8^{1,3}$ All others are planar.*

Proof. Let us consider a language L whose minimal automaton $\mathtt{A}_{min}$ has exactly 7 states. We can suppose that $\mathtt{A}_{min}$ does not contain any loop: 
with $x, y, z \in \Sigma$. Indeed, if such a case occurs, we perform the following replacement within the minimal automaton :



by


thus adding to the alphabet k new letters $X_1, \dots, X_k$. Let us call this new automaton A'. It corresponds to some regular language L' which is planar since its minimal automaton contains only 6 states. Let A^* be a corresponding planar automaton. We can replace in A^* any transition $r \xrightarrow{X_i} p'$ with p' in the equivalence class of p by $r \xrightarrow{x_i} q' \xrightarrow{y} p'$ with q' some freshly created state. The transformation is done without changing the genus of the underlying graph. Thus the result.

Thanks to Theorem 8, we enumerated all graphs G without loops such that any node v has exactly two out-going edges. To compute a direct emulator of some graph G_0, we used two algorithm.

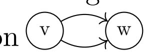
First, we have implemented a greedy algorithm `fast_is_planar` which can be briefly explained as follows. We begin with a graph G made of single node v projected to some node v_0 in G_0 and we mark it. Then, at each step, we pick some marked node in G, say u. Let us suppose that u is projected to u_0 in G_0, for any successor t_0 of u_0,

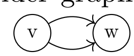
- either there is a node t in G such that adding an edge $v \to t$ in G keeps the graph planar, in which case we add the edge, or
- there is no such node t. Then, we add a new state w and a new edge $v \to w$ to G.

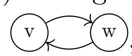
To keep the size of G under control, we use a global bound λ set to $21 = 3 \times 7$. The algorithm ends when there is no more marked node, in which case either we found a solution, and the answer is "Yes, it is planar", or not, that is "Don't know".

The implementation of the algorithm involves two lists, one for the collection of marked nodes, one for the collection of edges within G. Since we pick elements in these two lists, we may pick them either at the beginning or at the end (which more or less corresponds to some depth-first or breadth-first traversals). Since the algorithm is very fast, we try the four possibilities.

The other algorithm `full_is_planar` computes all planar graphs of size smaller than some limit which can be projected on G_0. There are finitely many of these, and thus the algorithm ends. Naturally, the complexity of this algorithm is much higher compared to the preceding one. Thus we apply the heuristics first. Within the 941 graphs without bigons, only 1 did not pass the first step. For all other graphs, the fast procedure was enough.

To avoid as most as possible the `X_is_planar` algorithms, we used the following additional trick. First, we verified that all graphs without bigons have a planar emulator. Consider a graph G with exactly one bigon 

between to nodes v and w. Suppose (A) that there is a node u without connections to v. Let G' be obtained from G by removing one of the edges between v and w. Consider now the graph G'' obtained from G' by adding an edge between v and u. It contains one bigon less than G, thus 0 bigon, thus has a planar emulator. Since G' is a subgraph of G'' so do G' and thus so do G. So, for graphs with one bigon, we don't need to consider graphs verifying (A). Avoiding (A) for a graph means it has the shape: 

For bigons of the shape 

, the approach is similar, and actually it ends on the same pattern. $\square$

Acknowlegments. This work was partially supported by the CIMI (Centre International de Mathématiques et d'Informatique), ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02 and by ANR-14-CE25-0005-3, ELICA.

References

[Bat62] Harary F., Kodama Y., Youngs J.W.T., Battle, J. Additivity of the genus of a graph. *Bull. Amer. Math. Soc.*, 68:565, 1962.

[BC76] Ronald V. Book and Ashok K. Chandra. Inherently nonplanar automata, *Acta Informat.* 6, no. 1, 89–94, 1976.

[BD13] Guillaume Bonfante and Florian Deloup. The genus of regular languages, *CoRR*, abs/1301.4981, 2013. To appear in Math. Structures in Computer Science, 2016.

[BP99] Ivona Bezáková and Martin Pál. Planar finite automata. Technical report, Student Science Conference, Comenius University, 1999.

[Eil74] Samuel Eilenberg. *Automata, Languages and Machines*, vol. A. Academic Press, New York 1974.

[Hli10] Petr Hlinený. 20 years of negami's planar cover conjecture. *Graphs and Combinatorics*, 26(4):525–536, 2010.

[Sak03] Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, Cambridge, 2009.

G. Bonfante, LORIA, Université de Lorraine, Campus scientifique, B.P. 239, 54506 Vandoeuvre-lès-Nancy Cedex guillaume.bonfante@loria.fr

F. Deloup, IMT, Université Paul Sabatier – Toulouse III, 118 Route de Narbonne, 31069 Toulouse cedex 9

florian.deloup@math.univ-toulouse.fr